

Copyright 2023 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM23-0557

# Symbolic Assurance Refinement

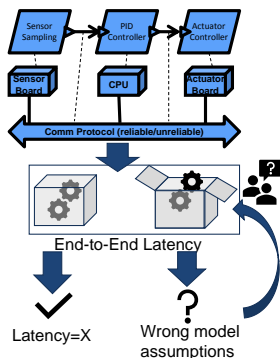
Dionisio de Niz and Lutz Wraga

Late Discovery of Design Errors in DoD Systems is very costly.

Architecture modeling and analysis can detect design error early

BUT:

- Analysis assumptions are often implicit
- if analysis assumptions not met: analyses break down for reasons not clear to users of analysis tools.
  - E.g., e2e Latency Assumption: periods multiple of each other (harmonic)



## SOLUTION

contract {

inputs:

E2ELatencies

assumptions:

areConnectionsDelayed()  
areDeadlinesConstrained()  
areTasksSchedulable()  
areAllThreadsPeriodic()

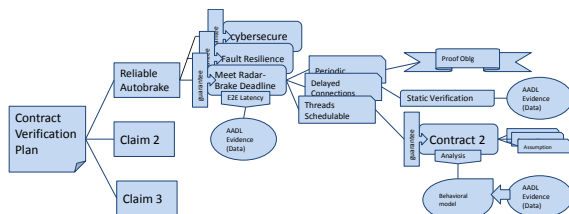
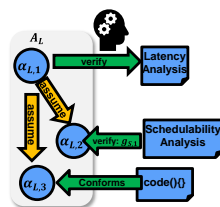
analysis:

meetEndToEndLatencies()

guarantee:

[E2EResponses[i] <= E2ELatencies[i]  
for i in range(len(Responses))]

$$C_L = (A_L, G_L) \text{ with } A_L = \{\alpha_{L,1}, \alpha_{L,2}, \alpha_{L,3}\}$$



**Symbolic Assurance:**

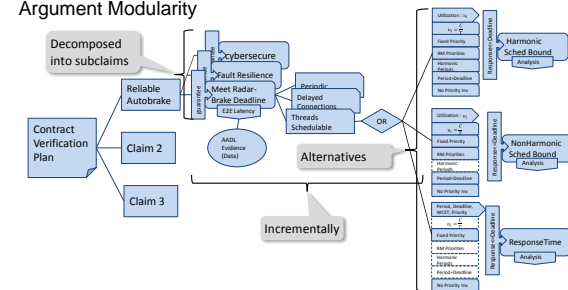
- Symbolic Satisfiability Problem Encoding (SMT)

**Argumentation evolves with model refinement**

- Unconstrained symbolic variables assigned value to satisfy problem
- Considered as proof obligations (due to partial model)

Verify final complete model for all possible values

## Argument Modularity



Algorithm 1 getSMTEncoding(F)()

```

1 F ← {f | (f, C) ∈ F from K}
2 T ← {C} | (f, C) ∈ F from K
3 while T ≠ ∅ do
4   select f from T and remove it from T
5   if f is argument then
6     T ← T ∪ {C}
7     F ← F ∪ (getSMTEncoding(f, C))
8   else if f is claim then
9     for p ∈ {p | (p, c) ∈ f} do
10      if p is contract then
11        T ← T ∪ p
12        F ← F ∪ (getSMTEncoding(p, c))
13      else
14        F ← F ∪ (p ⇒ c)
15    end if
16  end while
17 return F
  
```

**Refinement-Based Selection:**

- select contract based on model data availability

**Design-Based Selection:**

- Select contract based on model data type (e.g. RM priorities)